IN THE CLAIMS:

1.   (Original) A method in a data processing system for managing a set of memory resources used to store texture objects, the method comprising:

allocating memory to a current texture object in a first memory resource within the set of memory resources;

selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

repeating the allocating and selectively removing steps until the current texture object is allocated sufficient memory;

halting the repeating step in response to an absence of any stored texture objects being present in the first memory resource;

responsive to the halting step, selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

2.   (Original) The method of claim 1, wherein the first memory resource is system memory.

3.   (Original) The method of claim 1, wherein the first memory an advanced graphic port memory.

4.   (Original) The method of claim 1, wherein the second memory resource is a frame buffer.

5.   (Original) The method of claim 1, wherein the secondary memory resource is assigned to a kernel application.

6. (Original) The method of claim 1, wherein the first memory resource is assigned to a client application.

7. (Original) The method of claim 1, wherein the stored texture object is a texture object used less than a threshold value.

8. (Original) The method of claim 1, wherein the step of selectively removing texture objects comprises:

selectively removing all stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

9. (Original) The method of claim 1, wherein the step of selectively removing texture objects comprises:

selectively removing a single stored texture object in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

10. (Original) The method of claim 1, wherein an identifier is associated with a texture object when the texture object is removed from the set of memory resources in which the identifier identifies a memory resource in which the texture object was located within the set of memory resources.

11. (Currently amended) A memory management system comprising:

a memory allocation unit, wherein the memory allocation unit allocates memory to texture objects in a first memory resource and a second memory resource;

a texture management system, wherein the texture management system tracks allocation of memory for all texture objects, removes a stored texture object within the first memory resource in response to detecting an in-ability inability to allocate memory to a current texture object in the first memory resource, and calls the memory allocation unit to allocate memory to the current texture object after the stored texture object is removed, and continues to remove texture objects from the first memory resource until sufficient memory is allocated to the current texture object.

12.   (Original) The memory management system of claim 11, wherein the texture management system returns an error if stored texture objects are absent from the first memory resource and insufficient memory has been allocated to the current texture object.

13.   (Original) The memory management system of claim 11, wherein the texture management system removes texture objects from the second memory resource if texture objects are absent from the first memory resource and insufficient memory has been allocated to the current memory object.

14.   (Original) The memory management system of claim 11, wherein the texture management system includes a first texture manager and a second texture manager, wherein the first texture manager track allocation of memory for all texture objects in the first memory resource and wherein the second texture manager track allocation of memory for all texture objects in the second memory resource.

15.   (Original) A data processing system comprising:
        a bus system;
        a memory connected to the bus system, wherein a set of instructions are located in the memory; and
        a processor unit connected to the bus system, wherein the processor unit executes the set of instructions to allocate memory to a current texture object in a first memory resource within the set of memory resources; selectively remove a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object; repeat instructions to the allocate memory and selectively remove a stored texture object until the current texture object is allocated sufficient memory; halt the repeating in response to an absence of any stored texture objects being present in the first memory resource, selectively remove stored texture objects in a second memory resource in response to an inability to allocate sufficient memory to the current texture object in response to halting instructions to repeat; and allocate memory in the second

memory resource to the current texture object in response to selectively removing stored texture objects.

16. (Original) The data processing system of claim 15, wherein the bus system includes a primary bus and a secondary bus.

17. (Original) The data processing system of claim 15, wherein the processor unit includes a single processor.

18. (Original) The data processing system of claim 15, wherein the processor unit includes a plurality of processors.

19. (Cancelled)

20. (Original) A data processing system for managing a set of memory resources used to store texture objects, the data processing system comprising:

first allocating means for allocating memory to a current texture object in a first memory resource within the set of memory resources;

first selectively removing means for selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

repeating means for repeating initiation of the first allocating means and first selectively means removing steps until the current texture object is allocated sufficient memory;

halting means for halting the repeating step in response to an absence of any stored texture objects being present in the first memory resource;

second selectively removing means, responsive to the halting step, for selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

second allocating means for allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

21.   (Original) The data processing system of claim 20, wherein the first memory resource is system memory.

22.   (Original) The data processing system of claim 20, wherein the first memory an advanced graphic port memory.

23.   (Original) The data processing system of claim 20, wherein the second memory resource is a frame buffer.

24.   (Original) The data processing system of claim 20, wherein the secondary memory resource is assigned to a kernel application.

25.   (Original) The data processing system of claim 20, wherein the first memory resource is assigned to a client application.

26.   (Original) The data processing system of claim 20, wherein the stored texture object is a texture object used less than a threshold value.

27.   (Original) The data processing system of claim 20, wherein the step of selectively removing texture objects comprises:

    third selectively removing means for selectively removing all stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

28.   (Original) The data processing system of claim 20, wherein the step of selectively removing texture objects comprises:

    fourth selectively means for selectively removing a single stored texture object in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

29. (Original) The data processing system of claim 20, wherein an identifier is associated with a texture object when the texture object is removed from the set of memory resources in which the identifier identifies a memory resource in which the texture object was located within the set of memory resources.

30. (Original) A computer program product in a computer readable medium for managing a set of memory resources used to store texture objects, the computer program product comprising:

first instructions for allocating memory to a current texture object in a first memory resource within the set of memory resources;

second instructions for selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

third instructions for repeating the allocating and selectively removing steps until the current texture object is allocated sufficient memory;

fourth instructions for halting the repeating step in response to an absence of any stored texture objects being present in the first memory resource;

fifth instructions, responsive to the halting step, for selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

sixth instructions for allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.